

WHAT IS CLAIMED AND DESIRED TO BE SECURED BY LETTERS
PATENT OF THE UNITED STATES IS:

1. A method of slack allocation within a timing
5 graph, comprising the steps of:

setting an initial edge time for each edge in the
timing graph;

setting a weight value for each edge in the timing
graph; and

10 iteratively performing the following steps:

computing an amount of the available slack to be
budgeted to each edge of the graph;

for each edge having a weight greater than zero,

15 if the computed slack is greater than a predetermined
epsilon, then restore the edge time to that in an
immediately preceding iteration, and

if the computed slack is less than or equal to the
predetermined epsilon, then saving a current edge time and
setting a weight of the edge to zero;

20 stopping the iterative steps if an amount of slack to
allocate is less than the predetermined epsilon;

determining a new budget for each edge; and

incrementing the edge time for each edge by the
corresponding new budget ;

until the iterative steps are stopped or a maximum number of iterations is performed.

2. The method according to Claim 1, wherein said
5 step of determining a new budget comprises:

performing a timing cone analysis on domain
restricted timing cones within the timing graph to
determine a slack-to-weight ratio for each edge; and

10 increasing the edge time for each edge by an amount
equal to a weight of each edge multiplied by the
determined slack-to-weight ratio for that edge.

15 3. The method according to Claim 1, wherein said
timing graph is a set of vertices and edges representative
of timing of an electrical circuit.

4. The method according to Claim 1, wherein said
timing graph is a set of vertices and edges representative
of a model of a physical system.

20 5. A method for performing analysis on a set of
domain restricted timing cones, comprising the steps of:

selecting a domain restricted timing cone;

determining a slack to weight ratio for each edge in the domain restricted timing cone;

for each edge, if the determined slack to weight ratio for an edge is less than a current slack to weight
5 ration for that edge, then, replacing the current slack to weight ratio with the current slack to weight ratio;

selecting a next domain restricted timing cone and repeating said steps of determining and replacing until each timing cone has been analyzed.

6. The method according to Claim 5, further comprising the step of initially setting each slack to weight ration to a large value.

7. The method according to Claim 6, wherein said large value is infinity.

8. The method according to Claim 5, further comprising the step of excluding timing domains having
20 exception from analysis until the non excepted timing domains have been analyzed.

9. A method of computing a time budget for each edge in a Domain Restricted Timing Cone (DRTC) that can be represented in a graph, comprising the steps of:

5 determining timing constraints including a required output time (R) for an endpoint of the DRTC and an arrival time (A_i) for each input (i) of the DRTC;

decorating each vertex in the graph with a time to endpoint (tte) value and a weight to endpoint (wte) value;

10 computing a smallest slack to weight on each edge of the graph within the DRTC using the tte and wte pair for each edge.

15 10. The method according to claim 9, wherein tte represents a longest time to reach the endpoint and wte represents a largest sum of weight to reach the endpoint.

11. The method according to Claim 9, wherein said step of decorating each vertex comprises:

20 traversing the graph in the DRTC in a breadth first search (bfs).

12. The method according to Claim 9, wherein said step of computing a smallest slack to weight on each edge comprises:

traversing the graph in the DRTC in a breadth first search (bfs) and computing a slack to weight ratio for each edge.

5 13. The method according to Claim 9, wherein said arrival time (Ai) is computed according to:

 setting new Ai = Ai (edge_source) + edge_weight (edge) * slack_to_weight (edge); and

 if Ai(new) is greater than Ai (edge_sink), then
10 setting Ai(edge_sink)= new Ai.

 14. The method according to Claim 12, wherein:

 said slack to weight ratio for each edge is
calculated by computing a remaining_slack (edge) divided
15 by a remaining weight (edge);

 remaining_slack = remaining time (edge_source) - tte
(edge sink) - edge_time (edge);

 remaining_slack(edge)=remaining_time(edge source) -
tte (edge sink)-edge_time(edge);

20 remaining-weight (edge) = wte(edge sink) +
edge_weight(edge); and

 remaining_time(edge source) = R - an
arrival_time(edge source).

15. The method according to Claim 9, wherein said step of determining timing constraints comprises:

setting the output time at an arbitrary value; and

choosing an arrival time (A_i) for each input such

that $R - A_i$ is equal to a most constraining constraint imposed between the input A_i and the endpoint.

16. The method according to Claim 9, wherein:

said step of decorating comprises decorating each vertex in the graph with a set of time to endpoint (tte) and weight to endpoint (wte) pairs representing a time to reach and the endpoint and weight to reach the endpoint for each path from the vertex to the endpoint; and

said step of computing comprises computing a smallest slack to weight on each edge of the graph within the DRTC using the tte and wte pairs.

17. The method according to Claim 16, further comprising the step of:

removing wte and tte pairs that are not part of a smallest slack to weight ratio for any path from the vertex to the endpoint.

18. The method according to Claim 17, wherein:
said step of removing wte and tte pairs comprises,
setting an upper and lower bounds representing a
valid range for comparing wte and tte pairs, and
5 removing wte and tte pairs not having a lowest value
within said upper and lower bounds.

19. The method according to Claim 18, wherein said
step of removing wte and tte pairs further comprises
10 plotting a graph of each wte and tte pair on a slack to
weight vs remaining time grid.

20. The method according to Claim 17, wherein:
said step of removing wte and tte pairs comprises,
15 setting an upper and lower bounds representing a
valid range for comparing wte and tte pairs,

finding a first wte and tte pair having a lowest
slack to weight within the upper and lower bounds;

20 finding a second wte and tte pair having a lowest
slack to weight within the upper and lower bounds;

determining an intersection point (M) between the
first and second wte and tte pairs; and

comparing the remaining wte and tte pairs to said first wte and tte pairs between L and M, and to said second wte and tte pair between M and U.

5

21. The method according to Claim 17, wherein:

said step of removing wte and tte pairs comprises,

setting an upper and lower bounds representing a valid range for comparing wte and tte pairs,

finding a set of wte and tte pairs having a lowest slack to weight within the upper and lower bounds;

determining intersection points (M1, M2, ... Mn) between at a lowest slack to weight intersection of the lowest slack to weight wte and tte pairs;

removing the remaining wte and tte pair if the slack to weight ratio is not lower than the slack to weight ratio of any of the lowest slack to weight pairs between intersection points of other lowest slack to weight pairs representing a boundary of lowest slack to weight values between and said upper and lower bounds.

20

22. The method according to Claim 21, wherein:

said step of removing wte and tte pairs further comprises the step of adding remaining wte and tte pairs to the set of lowest slack to weight pairs if the

remaining wte and tte pair has a lower slack to weight ratio compared to any of the lowest slack to weight pairs between intersection points of other lowest slack to weight pairs and said upper and lower bounds.

5

23. The method according to Claim 22, further comprising the steps of:

modeling an equation using the lowest slack to weight pairs that represents a boundary of lowest slack to weight values between and said upper and lower bounds; and

using the modeled equation to determine if remaining slack to weight pairs should be included in the set of lowest slack to weight pairs.

10
15

24. The method according to Claim 22, further comprising the steps of:

if the number of slack to weight pairs to evaluate reaches a predetermined threshold, then,

establishing an envelope comprising a limited number of segments that follow a terrain of the lowest slack to weight pairs; and

20

rejecting any additional slack to weight pairs if the slack to weight of the additional pair is outside the established envelope.

25. The method according to Claim 9, wherein:

said method is embodied in a set of computer instructions stored on a computer readable media;

5 said computer instructions, when loaded into a computer, cause the computer to perform the steps of said method.

26. The method according to Claim 17, wherein said computer instruction are compiled computer instructions stored as an executable program on said computer readable media.

27. The method according to Claim 9, wherein said method is embodied in a set of computer readable instructions stored in an electronic signal.

28. The method according to claim 9, wherein tte represents a shortest time to reach the endpoint and wte represents a largest sum of weight to reach the endpoint.

29. The method according to Claim 9, wherein said step of computing a smallest slack to weight comprises:

maintaining a list of (wte, tte) pairs;

determining a lower (L) bounds and upper (U) bounds of timing to evaluate the pairs;

determining intersection points (M1, M2, ... Mn) between at a lowest slack to weight intersection of the lowest slack to weight wte and tte pairs;

removing pairs that do not lead to a smallest absolute value slack to weight ratio; and

computing the smallest slack to weight ration based on the remaining pairs.

30. The method according to Claim 29, wherein said (wte, tte) pairs, L and U, and intersection points are for an early case.

31. The method according to Claim 29, wherein said (wte, tte) pairs, L and U, and intersection points are for a negative delay allocation case.